EFFICIENT SPACETIME MESHING WITH NONLOCAL CONE CONSTRAINTS

Shripad Thite

Department of Computer Science, University of Illinois at Urbana-Champaign; thite@uiuc.edu

ABSTRACT

Spacetime Discontinuous Galerkin (DG) methods are used to solve hyperbolic PDEs describing wavelike physical phenomena. When the PDEs are nonlinear, the speed of propagation of the phenomena, called the *wavespeed*, at any point in the spacetime domain is computed as part of the solution. We give an advancing front algorithm to construct a simplicial mesh of the spacetime domain suitable for DG solutions. Given a simplicial mesh of a bounded linear or planar space domain M, we incrementally construct a mesh of the spacetime domain $M \times [0, \infty)$ such that the solution can be computed in constant time per element. We add a *patch* of spacetime elements to the mesh at every step. The boundary of every patch is *causal* which means that the elements in the patch can be solved immediately and that the patches in the mesh are partially ordered by dependence. The elements in a single patch are coupled because they share implicit faces; however, the number of elements in each patch is bounded. The main contribution of this paper is sufficient constraints on the progress in time made by the algorithm at each step which guarantee that a new patch with causal boundary can be added to the mesh at every step even when the wavespeed is increasing discontinuously. Our algorithm adapts to the local gradation of the space mesh as well as the wavespeed that most constrains progress at each step. Previous algorithms have been restricted at each step by the maximum wavespeed throughout the entire spacetime domain.

Keywords: mesh generation, unstructured meshes, advancing front, partial differential equations, discontinuous Galerkin, nonlinear hyperbolic PDE

1. INTRODUCTION

Simulation problems in mechanics consider the behavior of an object or region of space over time. Scientists and engineers use conservation laws and hyperbolic partial differential equations (PDEs) to model transient, wavelike phenomena propagating over time through the domain of interest. Example applications are numerous, including, for instance, the equations of elastodynamics in seismic analysis and the Euler equations for compressible gas dynamics. Closed-form solutions are typically unavailable for these problems, so analysts usually resort to numerical approximations.

Finite element methods (FEM) are popular options for solving this class of problems. In the standard *semi-discrete* approach, a finite element mesh discretizes space to generate a system of ordinary differential equations in time that is then solved by a time-marching integration scheme. Most semi-discrete methods impose a uniform time step size over the entire spatial domain, i.e., the time step does not adapt to the local gradation of the space mesh. Therefore, the resulting spacetime mesh consists of many more elements than required by physical causality. Hence, algorithms that use a nonuniform time step size can substantially improve computational efficiency.

Spacetime discontinuous Galerkin (DG) methods have been proposed by Richter [8], Lowrie *et al.* [7], and Yin *et al.* [11] for solving systems of nonlinear hyperbolic partial differential equations. Like traditional finite element methods, spacetime DG methods use basis polynomials to approximate the solution within each element; however, unlike traditional FEM methods, these basis polynomials have local support restricted to each element and the basis polynomials of adjacent elements do not have to agree on their common intersection. This approach eliminates artificial coupling between adjacent elements when the mesh satisfies certain causality constraints. (For further background on general discontinuous Galerkin methods, we refer the reader to Cockburn, Karniadakis, and Shu [3].)

Üngör and Sheffer [10] and Erickson *et al.* [4] developed the first algorithm, called 'TentPitcher', to build graded spacetime meshes over arbitrary simplicially meshed spatial domains, suitable for spacetime DG solutions. Unlike most traditional approaches, the TentPitcher algorithm does not impose a fixed global time step on the mesh, or even a local time step on small regions of the mesh. Rather, it produces a fully unstructured simplicial spacetime mesh, where the duration of each spacetime element depends on the local feature size and quality of the underlying space mesh.

Efficient spacetime meshing relies on the notion of the domain of influence and the domain of dependence of an event. Imagine dropping a pebble into a pond—circular waves propagate outwards from the point of impact. The frontier of expanding waves sweeps out a cone in spacetime called the domain of influence of the event. The radius of the domain of influence at any time is the radius of the circular disc consisting of all points on the surface where the initial wave has arrived. The domains of influence and dependence can be approximated by right circular cones with common apex P (Figure 1). The symmetric double cone representing the domains of influence and dependence at points P in spacetime can be described by a scalar field ω where $\omega(P) = \partial r/\partial t$, the *wavespeed* at P, specifies how quickly the radius r of domains of influence and dependence of P grows as a function of time. Smaller values of $\omega(P)$, i.e., steeper cones, correspond to slower wavespeeds. The wavespeed $\omega(P)$ at a point in spacetime is, in general, part of the solution of the PDE at that point. The *slope* of the cones of influence and dependence of P, denoted by $\sigma(P)$, is the reciprocal of the wavespeed—*larger slopes mean steeper cones and therefore slower wavespeeds, and smaller slopes mean shallower cones and faster wavespeeds.*

Given a simplicial mesh of some bounded domain $M \subset \mathbb{R}^d$, the Tent Pitcher algorithm incrementally constructs a simplicial mesh of the spacetime domain using an advancing front method. The spacetime domain is the subset $M \times [0, \infty) \subset \mathbb{R}^{d+1}$, a subset of Euclidean space one dimension higher. The algorithm progresses by adding simplices to the evolving mesh in small patches by moving a vertex of the front forward in time. The inflow and outflow boundaries of each patch (Figure 2) are *causal* by construction, i.e., each boundary facet F separates the cone of influence from the cone of dependence of any point on F (Figure 1). Equivalently, for every point P on F we have $\|\nabla F\| \leq 1/\omega(P) = \sigma(P)$. If the outflow boundaries of a patch are causal, every point in the patch depends only on other points in the patch or points of inflow elements adjacent to the inflow boundaries of the patch. Therefore, the solution within the patch can be computed as soon as the patch is created, given only the inflow data from adjacent inflow elements. The elements within a patch are causally dependent on each other and must be solved as a coupled system. Provided the space mesh has constant degree, each patch contains only a constant number of elements and can therefore be solved in constant time. Therefore, the computation time required to compute the numerical solution is linear in the number of spacetime elements. Patches with no causal relationship can be solved independently. To minimize undesirable numerical dissipation and the number of patches, we would like the boundary facets of each patch to be as close as possible to the causality constraint without violating it.

The causality constraint limits the progress in time at each step, i.e., the height of each tentpole is constrained. For spatial domains of dimension $d \geq 2$, it is not trivial to guarantee that the advancing front algorithm can always make progress. We require that for any target time value T the algorithm will compute a mesh of the spacetime volume $M \times [0, T]$ and the solution everywhere in this volume in finitely many steps. The target time T is not known *a priori* because it depends on the evolving physics. The original Tent Pitcher algorithm proposed by Üngör and Sheffer [10] applied to one- and two-dimensional space domains. The algorithm could guarantee progress only if the input triangulation contained only angles less than 90 degrees and if the wavespeed did not increase or increased smoothly. Erickson *et al.* [4] extended Tent Pitcher to arbitrary spatial domains in any dimensions by imposing additional constraints, called *progress constraints.* The progress constraint applied to a single simplex on the front limits the amount of progress in time when some vertex of the simplex is pitched. The progress constraint is a function of the shape of the simplex. The geometric constraints that limit the height of each tentpole are called *cone constraints.*

All the results so far have applied to the case where the wavespeed at a given point is either constant, decreasing, or increasing smoothly as a Lipschitz function. (See Alper Üngör's PhD thesis [9] for the details.) When the wavespeed changes, the previous algorithms take the fastest that the wavespeed can ever be and use that as a conservative upper bound on the wavespeed at any time. One would like an algorithm that adapts to increasing wavespeeds so that fewer spacetime elements, and therefore less computation time, are required to mesh a given volume.

In this paper, we give an advancing front algorithm to construct a spacetime mesh over an arbitrary linear or planar space mesh ($d \leq 2$). Our algorithm extends TentPitcher to the case when the wavespeed can be an arbitrary scalar field over the spacetime domain. In particular, our algorithm guarantees finite positive progress at each step even when the wavespeed at a given point increases discontinuously and unpredictably over time.

The main contributions of this paper are twofold. We give a novel characterization of fronts that are always guaranteed to progress, which we call *progressive fronts*, and give a lower bound on the progress guarantee at each step which depends only on the local size of the mesh and the wavespeed that most constrains the duration of the current patch. The minimum progress guarantee at any step is a positive quantity bounded away from zero, so the front is guaranteed to progress past any target time in a finite number of steps. The second contribution of this paper is to give geometric constraints on the front at any step that guarantee that the front can progress in the next step and so on inductively at every step. The geometric constraints are simple to express and to compute. Intuitively, the geometric constraints that apply at any given iteration of the algorithm are predicted by looking ahead at the next iteration of the algorithm. We also give an efficient algorithm to maximize the progress at every step subject to these constraints. The novelty of our characterization of progressive fronts and of our algorithm is that we resolve the following *conundrum*. The progress of the front at each step i is limited by the progress constraint that must be satisfied by the next front at step $i+1$. However, we do not know what is the next front unless we know how much progress is possible at step i.

The paper by Erickson *et al.* [4] contains an error in the statement of the causality constraint when obtuse triangles are involved; therefore, their proof of correctness is incomplete because it omits the obtuse angle case. While their proof can be fixed, we prefer our new algorithm, which is provably correct even when the wavespeed is constant or does not increase. Our new progress constraints are potentially weaker than those of Erickson *et al.* [4].

Our algorithm is the first algorithm to build spacetime meshes over arbitrary planar triangulated spatial domains suitable for solving nonlinear hyperbolic PDEs, where the wavespeed at any point in spacetime depends on the solution and cannot be computed in advance. Moreover, the solution can change discontinuously, for instance when a *shock* propagates through the domain.

The input to our advancing front algorithm is a simplicially meshed bounded domain $M \subset \mathbb{R}^d$ where $d \leq 2$ and the initial conditions of a nonlinear hyperbolic PDE. The space mesh describes the situation at time equal to zero, specifically, the slope at every point in M at time zero. We allow more general initial conditions but we will postpone a description of those conditions until later sections. Our meshing algorithm is an advancing front procedure which alternately constructs a new patch of elements and invokes a spacetime DG finite element method to compute the solution within that patch. At every iteration, the *front* is the graph of a continuous piecewise linear time function $t : M \to \mathbb{R}$. The front t is linear within every simplex of M and $\|\nabla t(p)\| \leq \sigma(p)$ for every point $p \in M$. The front is a terrain whose facets correspond to simplices in the underlying space mesh. Each facet of the front coincides with the outflow face of a patch in the past and the inflow face of a patch in the future. We say that a front is *causal* if every simplex of the front is causal. To advance the front t, the algorithm chooses an arbitrary vertex $P = (p, t(p))$ from the front and lifts it to a new point $P' = (p, t'(p))$ where $t'(p) > t(p)$ and for every other vertex q we have $t'(q) = t(q)$. The spacetime volume between the new front t' and the old front t is called a *tent*. The tent is meshed with simplices sharing the edge (P, P') called the *tentpole*. The *height* of the tentpole is the duration $t'(p) - t(p)$. Consider a planar space mesh M. For each triangle pqr incident on p, the tetrahedron $P'PQR$ belongs to the patch. The outflow face $P'QR$ and the inflow face PQR are causal boundaries. The triangles $P'PQ$ and $P'PR$ are implicit faces. Since the implicit faces are vertical they

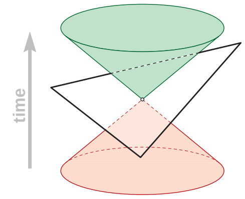

Figure 1: A causal face separates the cones of influence and dependence at every point on the face.

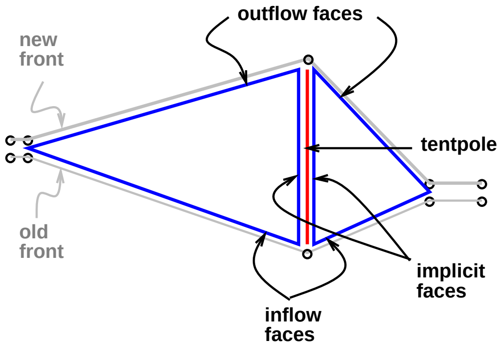

Figure 2: A vertical cross-section of a patch of tetrahedra; the inflow and outflow faces are causal.

are not causal boundaries and so elements within the patch are coupled. The elements below the front t whose outflow faces intersect any of the inflow faces of the new patch are inflow elements. We pass the newly constructed patch along with all its inflow elements to a DG solver. The DG solver returns as part of the solution the slope at every point on every outflow face of the patch. The new front t' and the output of the DG solver are the input to the next iteration of the algorithm.

Since we are interested in causal fronts only, henceforth it is implicit that every front considered is causal.

We assume that the slope at any point P is bounded by the minimum and the maximum slopes anywhere in the cone of dependence of P. Hence, given a front t and a point P in the future, the slope at P is no smaller than the slope at Q for every point Q on the front t such that P is in the cone of influence of Q.

It can be computationally very expensive to determine the shallowest cone of influence that contains a given point P. In particular, the shallowest cone of influence containing P may correspond to a *nonlocal* point Q, one arbitrarily distant from P. To compute this nonlocal cone constraint efficiently, we use a standard hierarchical decomposition, called a *bounding cone hierarchy*, of the space domain. The elements in the hierarchy correspond to subsets of the space domain. For each element of the hierarchy, we compute the minimum slope within the corresponding subset of the space domain. The smallest element in the hierarchy is a single simplex. In order to determine the strictest cone constraint that applies locally, we traverse the hierarchy until we determine the simplex with minimum slope whose cone of influence contains P. In practice, we expect that our algorithm has to examine only a small subset of the hierarchy. In the worst case, the algorithm has to examine every simplex of the front but in that case the algorithm will be at most a constant factor slower than one that does not use a bounding cone hierarchy. When a patch is solved, the bounding cones are updated with the new slopes by traversing a path from a leaf to the root of the hierarchy. This hierarchical approximation technique has been applied very successfully to numerous simulation problems, such as the Barnes-Hut divide-and-conquer method [2] for N-body simulations, as well as to collision detection in computer graphics and robot motion planning [6] and for indexing multi-dimensional data in geographic information systems [5].

1.1 Notation

We use lowercase letters like p, q, r to denote points in space and uppercase letters like P, Q, R to denote points in spacetime. A front t is a piecewise linear function $t : M \to \mathbb{R}$. For a simplex (of any dimensions) τ of M, let $t|_\tau$ denote the time function t restricted to τ and extended to the affine hull of τ; in other words, $t|_\tau$ is a linear function that coincides with t for every point of τ. Let $t_i : M \to \mathbb{R}$ denote the front after the ith step of the algorithm; t_0 is the initial front. For every i, the front t_i is a terrain whose facets are the simplices of M. In other words, t_i is a piecewise linear function such that for every simplex τ of M, the functions t_i and $t_i|_\tau$ coincide at the vertices of τ.

For a time function $t : M \to \mathbb{R}$ we denote the gradient of t by ∇t. A local minimum of the front t is a vertex p such that $t(p) \le t(q)$ for every vertex q that is a neighbor of p. When the current front t is clear from the context, for every point $p \in M$ we use P to denote the corresponding point on the front, i.e., $P = (p, t(p))$.

For a point P in spacetime, we use $\sigma(P)$ to denote the reciprocal of the wavespeed at P. Let $\sigma_{\min}$ denote $\min_{P \in M \times [0,\infty)}\{\sigma(P)\}$ and $\sigma_{\max}$ denote $\max_{P \in M \times [0,\infty)}\{\sigma(P)\}$. We assume that $0 < \sigma_{\min} \le \sigma_{\max} < \infty$. For a simplex τ in spacetime, we use $\sigma(\tau)$ to denote the minimum of $\sigma(P)$ over all points P in τ.

We say that a front t' is obtained by advancing a vertex p of M by $\delta t \ge 0$ if $t'(p) = t(p) + \delta t$ and for every other vertex $q \ne p$ we have $t'(q) = t(q)$. For any front t, vertex p, and real $\delta t \ge 0$, let $t' = \text{next}(t, p, \delta t)$ denote the front obtained from t by advancing p by δt.

1.2 Problem statement

The input to our problem is the initial front t_0 and the initial conditions of the PDE. We want an advancing front algorithm such that for every $T \in \mathbb{R}^{\ge 0}$ there exists a finite integer $k \ge 0$ such that the front t_k after the kth iteration of the algorithm satisfies $t_k \ge T$.

We say that a front t is *valid* if there exists a positive real δ bounded away from zero such that for every $T \in \mathbb{R}^{\ge 0}$ there exists a sequence of fronts t, t_1, t_2, …, t_k where $t_k \ge T$, each front in the sequence obtained from the previous front by advancing some vertex by δ. What makes the definition of a valid front nontrivial is the requirement that all fronts be causal. The main difficulty in characterizing valid fronts arises when the wavespeed at a given point in the space domain increases discontinuously and unpredictably over time.

Our solution We define *progressive* fronts and prove that if a front is progressive then it is valid. We give an algorithm that given any progressive front t_i constructs a next front t_{i+1} such that t_{i+1} is progressive. The volume between t_i and t_{i+1} is partitioned into simplices. The next front t_{i+1} is obtained by lifting a local minimum of t_i by a positive amount bounded away from zero. The algorithm can easily be parallelized to solve several patches asynchronously by lifting any independent set of vertices in parallel. Whenever the algorithm chooses to lift a local minimum, it is guaranteed to be able to lift it by at least $T_{\min} > 0$ which is a function of the input and bounded away from zero.

2. ONE-DIMENSIONAL SPACE DOMAINS

We begin by describing our algorithm to construct spacetime meshes over one-dimensional space domains. Even this simple case captures all but one aspect of the complexity of guaranteeing causality when wavespeeds are changing.

The space domain M is a closed interval of the real line. The input space mesh is a subdivision of this interval into segments. Let $V(M)$ denote the set of vertices of the space mesh M. The initial front t_0 corresponds to $t_0(p) = 0$ for every vertex p of the space mesh, but more generally, any (causal) front can be the initial front. Let $w_{\min}$ denote the minimum length of any segment in the space mesh. Let $\sigma_{\min}$ denote the minimum slope $\sigma(P)$ over every point P in the spacetime domain $M \times [0, \infty)$. Let $T_{\min}$ denote $\sigma_{\min} w_{\min}$.

In iteration $i + 1$ of our advancing front algorithm ($i \geq 0$), we advance a single vertex p, where p is a local minimum of the current front t_i, to get the new front t_{i+1}, i.e., $t_{i+1} = \text{next}(t, p, \delta t)$. More generally, we can advance any vertex or an independent set of vertices, not necessarily local minima, forward in time. The value of $t_{i+1}(p)$ is bounded from above by the requirement that t_{i+1} be causal.

Let AB be an arbitrary segment of the front t_{i+1}. Without loss of generality, assume $t_{i+1}(a) \leq t_{i+1}(b)$. Then, AB is causal if and only if the gradient of the time function t_{i+1} restricted to ab is at most the slope $\sigma(AB)$, i.e., if and only if

$$\boxed{\|\nabla\ t_{i+1}|_{ab}\| = \frac{t_{i+1}(b) - t_{i+1}(a)}{|ab|} \leq \sigma(AB).} \tag{1}$$

Theorem 1. *Let t_i be a front and let p be an arbitrary local minimum of t_i. Then, for every $\delta t \in [0, T_{min}]$ the front $t_{i+1} = next(t_i, p, \delta t)$ is causal.*

Proof. Only the segments of the front incident on P advance along with p. Consider an arbitrary segment pq incident on p. Let t and t' denote $t_i|_{pq}$ and $t_{i+1}|_{pq}$ respectively. We have $t(p) + \delta t \leq t(p) + \sigma_{\min} w_{\min} \leq t(q) + |pq| \sigma(P'Q)$ because p is a local minimum, $w_{\min} \leq |pq|$, and $\sigma_{\min} \leq \sigma(P'Q)$. Therefore, the segment $P'Q$ is causal. Since this is true of an arbitrary segment on the front t', we have proved that that the front $t_{i+1} = \text{next}(t_i, p, \delta t)$ is causal. □

Theorem 2. *For any $i \geq 0$, if the front t_i is causal then t_i is valid.*

Proof. Consider step $i + 1$ of the algorithm. By Theorem 1 the front t_{i+1} such that $t_{i+1}(p) \in [0, T_{\min}]$ is causal. Therefore, we have shown that if t_i is causal then there is a front $t_{i+1} = \text{next}(t_i, p, T_{\min})$ such that t_{i+1} is causal. Note that $\sum_{p \in V(M)} t_{i+1}(p) = T_{\min} + \sum_{p \in V(M)} t_i(p)$. By induction on i, and because $\sigma_{\max}$ is finite and M is bounded, there exists a finite $k \geq i$ such that the front t_k satisfies

$$\sum_{p \in V(M)} t_k(p) \geq \text{diam}(M) \sigma_{\max} T$$

for any real T. Since t_k is causal

$$\left(\max_{p \in V(M)} t_k(p) \right) \leq \text{diam}(M) \sigma_{\max} \left(\min_{p \in V(M)} t_k(p) \right).$$

Therefore, $\min_{p \in V(M)} t_k(p) \geq T$ and so t_i is valid. □

2.1 Being greedy at every step

We would like to maximize the progress at each step in a greedy fashion, i.e., given a front t_i we would like to maximize $t_{i+1}(p)$ where $t_{i+1} = \text{next}(t_i, p, \delta t)$ subject to the constraint that t_{i+1} is causal. By Theorem 2, we can have $t_{i+1}(p) \geq t_i(p) + T_{\min}$. However, it may be possible to make further progress by setting $t_{i+1}(p)$ higher, especially if each segment PQ incident on p each satisfies progress constraint $[\sigma_{\text{prev}}]$ for some $\sigma_{\text{prev}} < \sigma(P'Q)$ at the end of the previous iteration.

For a fixed segment pq incident on p let $T_{\sup}^{i+1}$ denote $\sup\{T : P'Q \text{ is causal where } P' = (p, T)\}$. To maximizing the progress at step $i + 1$, we would like to compute $T_{\sup}^{i+1}$. The segment $P'Q$ is causal if and only if the slope of $P'Q$ is less than or equal to the slope of the cone of influence from every point on the front that intersects $P'Q$. A cone of influence intersects $P'Q$ if and only if the cone intersects the tentpole PP'. In general, a cone of influence from arbitrarily far away can intersect the tentpole at p. See Figure 3. This is not the case when the wavespeed everywhere is the same. Therefore, in general, $T_{\sup}^{i+1}$ could be determined by a cone of influence of a point arbitrarily distant from p.

Partition the front into two subsets of points: (i) points in the star of P ("local" points), and (ii) points everywhere else on the front ("remote" points). Corresponding to each subset we have two disjoint subsets of cones of influence—$\mathcal{C}_{\text{local}}$ and $\mathcal{C}_{\text{remote}}$ respectively. Each subset of cones limits the new time value of p and so the final time value is the smaller of the two values for each of $\mathcal{C}_{\text{local}}$ and $\mathcal{C}_{\text{remote}}$ taken separately.

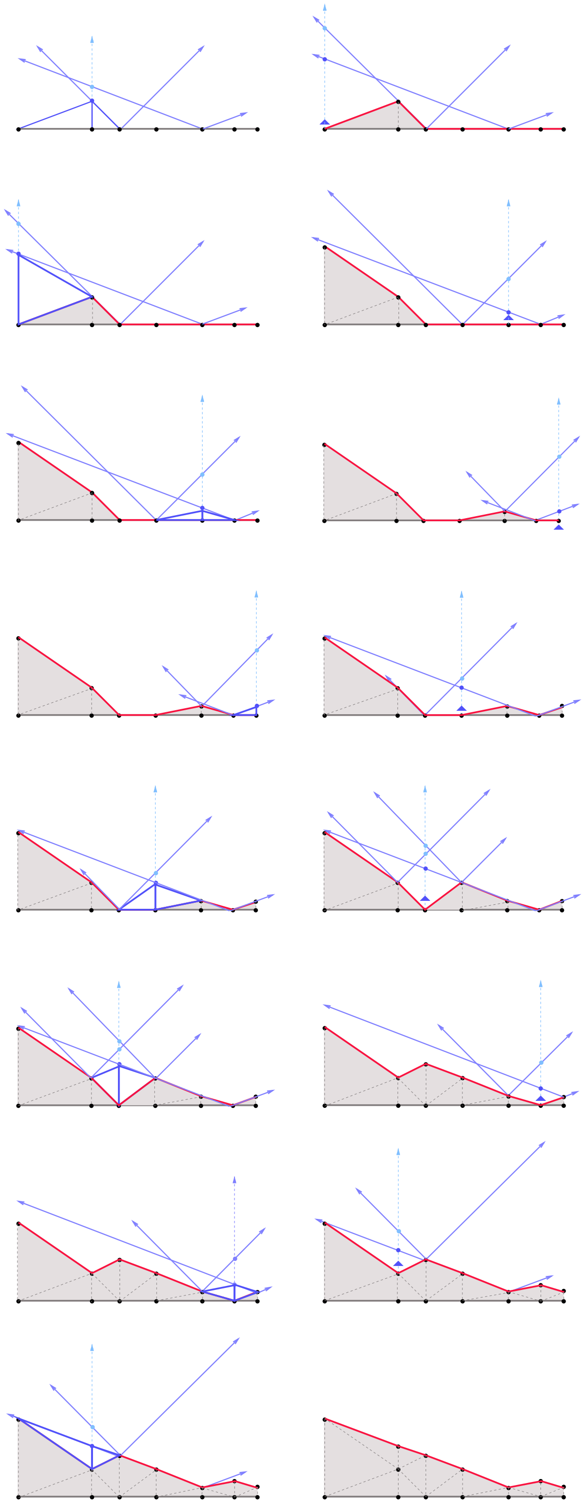

Figure 3: Top to bottom: a sequence of tent pitching steps in 1D×Time. Maximizing the height of each tentpole while staying below every cone of influence can require examining remote cones arbitrarily far away.

Consider the subset $\mathcal{C}_{\text{local}}$. Let σ_{local} denote the smallest slope among all cones of influence in $\mathcal{C}_{\text{local}}$ The segment $P'Q$ is causal only if its slope is less than or equal to σ_{local}. Let T_{local} be the maximum time value of P' for which the slope of $P'Q$ is less than or equal to σ_{local}. The maximum T_{local} exists because the set of feasible values is closed and therefore compact. To compute T_{local} we substitute σ_{local} in the condition for causality of $P'Q$ (Equation 1).

Next consider the subset $\mathcal{C}_{\text{remote}}$. The front t_i is strictly below every cone in $\mathcal{C}_{\text{remote}}$ because t_i is causal. The segment $P'Q$ is causal only if it is also strictly below every cone in $\mathcal{C}_{\text{remote}}$. Given a cone $C \in \mathcal{C}_{\text{remote}}$, C intersects $P'Q$ if and only if C intersects the tentpole PP'. Let T_{remote} denote the smallest time value T for which the tentpole PP' where $P' = (p, T)$ intersects exactly one cone in $\mathcal{C}_{\text{remote}}$. The segment $P'Q$ is causal only if $T < T_{\text{remote}}$. Note that the upper bound on T imposed by remote cones is a strict inequality.

Therefore, the progress $t_{i+1}(p) - t_i(p)$ at step $i+1$ is limited because $T_{\text{sup}}^{i+1} = \max\{T_{\text{local}}, T_{\text{remote}}\}$. To maximize the progress at the current step, we choose $t_{i+1}(p)$ equal to T_{sup}^{i+1} minus the machine precision η, or $t_i(p) + T_{\min}$, whichever is larger.

Computing $\mathbf{T}_{\text{remote}}$ exactly Computing T_{remote} is equivalent to answering a ray shooting query in the arrangement of the cones in $\mathcal{C}_{\text{remote}}$. We use a bounding cone hierarchy $\mathcal{H}$ obtained from a hierarchical decomposition of the space domain to efficiently answer the ray shooting query. The hierarchical decomposition of the space domain induces a corresponding hierarchical decomposition of every front. For each element of this hierarchy, we store a right circular cone that bounds the cone of influence of every point of the corresponding subset of the front. To answer the ray shooting query, we traverse the cone hierarchy from top to bottom starting at the root. At every stage, we store a subset $\mathcal{C}$ of bounding cones such that every cone in $\mathcal{C}_{\text{remote}}$ is contained in some cone in the subset $\mathcal{C}$. The cones in $\mathcal{C}$ are stored in a priority queue in non-decreasing order of the time value at which the vertical ray at P intersects each cone. Initially, $\mathcal{C}$ consists solely of the cone at the root of the hierarchy. At every stage, if the cone in $\mathcal{C}$ that has the earliest intersection does not come from a leaf in the hierarchy then we replace it in the priority queue with its children. Continuing in this fashion, we eventually determine the single facet of the front such that the cone of influence from some point on this facet is intersected first by the vertical ray at P. The time coordinate of the point of intersection is T_{remote}, the answer to the ray shooting query.

If the hierarchy is balanced its depth is $O(\log m)$ where m is the number of simplices in the space mesh. In

1D×Time, we observed empirically that on average only a few nodes in the cone hierarchy were examined by this algorithm to determine the most constraining cone of influence.

Approximating $\mathbf{T_{remote}}$ Since we know a range of values $[t_i(p)+T_{\min}, T_{\mathrm{local}}]$ that contains T_{remote}, we can approximate T_{remote} up to any desired numerical accuracy by performing a binary search in this interval. At every iteration, we speculatively lift P to the midpoint of the current search interval. Let P'' be the speculative top of the tentpole at P. We query the cones of influence in $\mathcal{C}_{\mathrm{remote}}$ to determine the minimum slope σ_{remote} among all cones that intersect PP''. If the maximum slope of the outflow faces incident on P'' is less than σ_{remote} then we can continue searching in the top half of the current interval; otherwise, the binary search continues in the bottom half of the current interval. The search terminates when the search interval is smaller than our desired accuracy. A bounding cone hierarchy helps in the same manner as before to determine the minimum slope among all cones in $\mathcal{C}_{\mathrm{remote}}$ that intersect PP''.

Theorem 3. *Given a simplicial mesh M of a bounded real interval where w_{min} is the minimum length of a simplex of M and σ_{min} is the minimum slope anywhere in $M \times [0,\infty)$ our algorithm constructs a simplicial mesh of $M \times [0,T]$ consisting of at most $\left\lceil \frac{2\, diam(M)\, \sigma_{max}}{\sigma_{min} w_{min}} T \right\rceil$ spacetime elements for every real $T \geq 0$.*

Proof. In Theorem 1, we have shown that the height of each tentpole constructed by the algorithm is at least $T_{\min} = \sigma_{\min} w_{\min}$. By Theorem 2, after constructing at most $k \leq \left\lceil \frac{\mathrm{diam}(M)\, \sigma_{\max}}{T_{\min}} T \right\rceil$ patches, the entire front t_k is past the target time T. Since each patch consists of at most two elements, the theorem follows. □

We have shown that every causal front in 1D×Time is valid. In higher dimensions, additional progress constraints are necessary.

3. PLANAR SPACE DOMAINS

In this section, we describe our algorithm for $d = 2$, i.e., for a triangulated planar space domain $M \subset \mathbb{R}^2$.

For planar domains, we encounter nontrivial progress constraints that are necessary to guarantee sufficient progress at each step, i.e., to guarantee that the height of the tentpole constructed at every step is positive and bounded away from zero. In the absence of such constraints, it was shown by Üngör and Sheffer [10], and by Erickson *et al.* [4] that if the space mesh contains an obtuse or a right triangle then Tent Pitcher will eventually construct a front such that no further progress is possible while maintaining causality. Erickson *et al.* [4] derived additional progress constraints that were sufficient to guarantee progress, even in the presence of obtuse angles, however only by assuming the minimum slope occurs everywhere in spacetime. In this section, we show how to relax these progress constraints so that they adapt to the slope of the most constraining cone of influence at every step. Our progress constraint is a function of the slope encountered locally in the next step of the algorithm, which may be substantially less constraining than the globally minimum slope.

Fix a real parameter $\varepsilon \in \left(0, \frac{1}{2}\right]$. The space domain M is a triangulation of a bounded subset of the plane $\mathbb{R}^2$. Let $w_{\min}$ denote the minimum width of any triangle of the space mesh. Let $\sigma_{\min}$ denote the minimum $\sigma(P)$ over every point P in the spacetime domain $M \times [0, \infty)$. Let $T_{\min}$ denote $\varepsilon \sigma_{\min} w_{\min}$.

Definition 1 (Progress constraint σ)**.** *Let PQR be an arbitrary triangle of a front t. Without loss of generality, assume $t(p) \leq t(q) \leq t(r)$. We say that the triangle PQR satisfies* progress constraint σ *if and only if*

$$\|\nabla\, t|_{qr}\| := \frac{t(r) - t(q)}{|qr|} \leq (1 - \varepsilon)\sigma\phi_p$$

where $\phi_p = \max\{\sin\angle prq, \sin\angle pqr\}$. Note that $0 < \phi_p \leq 1$.

Suppose the lowest vertex p is being advanced. As long as p is the lowest vertex of $\triangle pqr$, the progress constraint limits $\|\nabla\, t|_{qr}\|$ but $\|\nabla\, t|_{qr}\|$ is unchanged by lifting p. When $t(p) > t(q)$, the new lowest vertex is q, so the progress constraint limits $\|\nabla\, t|_{rp}\|$. (We can interpret the progress constraint inductively as a causality constraint on the 1-dimensional facet pr opposite q where the relevant slope is $(1 - \varepsilon)\sigma\phi_q$.)

Definition 2 (Progressive)**.** *Let t be a front and let pqr be a given triangle. Without loss of generality, assume $t(p) \leq t(q) \leq t(r)$. We say that the triangle PQR is* progressive *if and only if both of the following conditions are satisfied by $P'QR$ where $P' = (p, t(p) + \delta t)$ for every $\delta t \in [0, T_{min}]$:*

1. *$P'QR$ is causal, and*

2. *$P'QR$ satisfies progress constraint $\sigma(P'Q'R)$ where $Q' = (q, t(q) + T_{min})$.*

We say that a front t is *progressive* if every triangle on the front is progressive. Note that every progressive triangle or front is also causal.

3.1 A new advancing front algorithm

We are now ready to describe iteration $i+1$ of our advancing front algorithm for $i \geq 0$. Advance a single vertex p by a positive amount, where p is any local minimum of the current front t_i, to get the new front t_{i+1} such that for every triangle pqr incident on p the corresponding triangle on the new front t_{i+1} is progressive. In the parallel setting, advance any independent set of local minima forward in time, each subject to the above constraint. The value of $t_{i+1}(p)$ is constrained from above separately for each of the simplices incident on p. The final value chosen by the algorithm must satisfy the constraints for each such triangle. Therefore, it is sufficient to consider each triangle pqr incident on p separately while deriving the causality and progress constraints that apply while pitching p.

Next, we derive simple formulæ for the causality and progress constraints for a given triangle pqr when p is being pitched. Let t and t' denote $t_i|_{pqr}$ and $t_{i+1}|_{pqr}$ respectively.

Let $\vec{n}_{qr}$ denote the unit vector normal to qr such that $\vec{n}_{qr} \cdot (\vec{p} - \vec{q}) > 0$. Let $\vec{v}_{qr}$ be the unit vector parallel to qr such that $\vec{v}_{qr} \cdot (\vec{r} - \vec{q}) > 0$. Then, $\{\vec{n}_{qr}, \vec{v}_{qr}\}$ form a basis for the vector space $\mathbb{R}^2$. Let $\vec{n}_{rp}$ denote the unit vector normal to pr such that $\vec{n}_{rp} \cdot (\vec{q} - \vec{p}) > 0$. Let $\vec{v}_{rp}$ be the unit vector parallel to rp such that $\vec{rp} \cdot (\vec{p} - \vec{r}) > 0$. Then, $\{\vec{n}_{rp}, \vec{v}_{rp}\}$ form another basis for the vector space $\mathbb{R}^2$.

The gradient vector $\nabla t'$ can be written as

$$\nabla t' = (\nabla t' \cdot \vec{n}_{qr})\vec{n}_{qr} + \nabla t'\big|_{qr}$$

where

$$\nabla t'\big|_{qr} = (\nabla t' \cdot \vec{v}_{qr})\vec{v}_{qr}$$

Lifting p does not change the gradient of the time function restricted to the opposite edge, so $\nabla t' \cdot \vec{v}_{qr} = \nabla t \cdot \vec{v}_{qr}$, i.e., $\nabla t'|_{qr} = \nabla t|_{qr}$. Since q is the lowest vertex of qr, we have $\nabla t' \cdot \vec{v}_{qr} = \nabla t \cdot \vec{v}_{qr} \geq 0$.

Also,

$$\nabla t' = (\nabla t' \cdot \vec{n}_{rp})\vec{n}_{rp} + \nabla t'\big|_{rp}$$

where

$$\nabla t'\big|_{rp} = (\nabla t' \cdot \vec{v}_{rp})\vec{v}_{rp}$$

The vectors $\vec{n}_{qr}$ and $\vec{n}_{rp}$ are related by a rotation around the origin by angle θ. Since $0 < \theta < \pi$ we have $\cos\theta = \vec{n}_{qr} \cdot \vec{n}_{rp}$ and $\sin\theta = \sqrt{1 - (\vec{n}_{qr} \cdot \vec{n}_{rp})^2}$. Hence,

$$\begin{aligned} \|\nabla t'\big|_{rp}\| &= \|\nabla t\big|_{qr}\| \cos\theta + (\nabla t' \cdot \vec{n}_{qr}) \sin\theta \\ &= \|\nabla t\big|_{qr}\|(\vec{n}_{qr} \cdot \vec{n}_{rp}) \\ &\quad + (\nabla t' \cdot \vec{n}_{qr})\sqrt{1 - (\vec{n}_{qr} \cdot \vec{n}_{rp})^2} \end{aligned} \qquad (2)$$

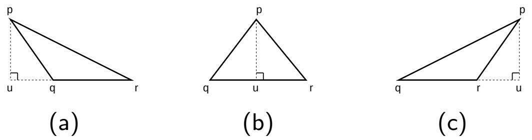

Figure 4: Triangle pqr where $t(p) \leq t(q) \leq t(r)$

Deriving the causality constraint Let u be the orthogonal projection of p onto line qr. Since lifting p does not change the time function restricted to qr, we have $t'|_{qr} = t|_{qr}$. The scalar product $\nabla t' \cdot \vec{n}_{qr}$ can be written as

$$\nabla t' \cdot \vec{n}_{qr} = \frac{t'(p) - t(u)}{|up|}$$

Since q is the lowest vertex of qr and since PQR is progressive, we have $0 \leq \nabla t' \cdot \vec{v}_{qr} = \nabla t \cdot \vec{v}_{qr} \leq (1-\varepsilon)\sigma(P'QR) < \sigma(P'QR)$. Therefore, $\|\nabla t'\| \leq \sigma(P'QR)$ if and only if

$$\boxed{\frac{t'(p) - t(u)}{|up|} \leq \sqrt{\sigma(P'QR)^2 - \|\nabla t|_{qr}\|^2}} \qquad (3)$$

Deriving the progress constraint Let σ_{prog} denote $\sigma(P'Q'R)$ where $P' = (p, t'(p))$ and $Q' = (q, t(q) + T_{\min})$. By Equation 2, the triangle $P'QR$ satisfies the progress constraint $\|\nabla t'|_{rp}\| \leq (1-\varepsilon)\sigma_{\text{prog}}\phi_q$ if and only if

$$\nabla t' \cdot \vec{n}_{qr} \leq \frac{(1-\varepsilon)\sigma_{\text{prog}}\phi_q - \|\nabla t|_{qr}\|(\vec{n}_{qr} \cdot \vec{n}_{rp})}{\sqrt{1 - (\vec{n}_{qr} \cdot \vec{n}_{rp})^2}}$$

Therefore, the progress constraint is

$$\boxed{\begin{aligned} \tfrac{t'(p)-t(u)}{|up|} \quad \leq \quad & \tfrac{1}{\sqrt{1-(\vec{n}_{qr}\cdot\vec{n}_{rp})^2}}\,(1-\varepsilon)\sigma(P'Q'R)\phi_q \\ & - \tfrac{\vec{n}_{qr}\cdot\vec{n}_{rp}}{\sqrt{1-(\vec{n}_{qr}\cdot\vec{n}_{rp})^2}}\,\|\nabla t|_{qr}\| \end{aligned}} \qquad (4)$$

3.2 Proof of correctness

In this section, we prove the correctness of our algorithm, i.e., that every front constructed by the algorithm is valid.

Theorem 4. *If a front t_i is progressive, then for any local minimum vertex p and for every $\delta t \in [0, T_{min}]$ the front $t_{i+1} = next(t_i, p, \delta t)$ is causal.*

Proof. Since only the triangles of the front incident on P advance along with p, we can restrict our attention to an arbitrary triangle pqr incident on p. Let t and t' denote $t_i|_{pqr}$ and $t_{i+1}|_{pqr}$ respectively. Let u be the orthogonal projection of p onto line qr.

Consider the causality constraint (Equation 3). We will consider two cases separately: (i) $t(u) \geq t(q)$, and (ii) $t(u) < t(q)$.

Case 1: $\mathbf{t(u) \geq t(q) \geq t(p)}$ See Figure 4(b)–(c). In this case, we have

$$\begin{aligned} t'(p) &= t(p) + \delta t \\ &\leq t(u) + \delta t \\ &\leq t(u) + \varepsilon \sigma_{\min} w_{\min} \\ &\leq t(u) + \varepsilon \sigma(P'QR)|up| \end{aligned}$$

because $|up| \geq w_{\min}$ and $\sigma(P'QR) \geq \sigma_{\min}$. Since $0 < \varepsilon \leq \frac{1}{2}$ we have $\varepsilon \leq \sqrt{1-(1-\varepsilon)^2}$. Therefore,

$$\begin{aligned} t'(p) &\leq t(u) + |up|\sqrt{1-(1-\varepsilon)^2}\sigma(P'QR) \\ &= t(u) + |up|\sqrt{\sigma(P'QR)^2 - (1-\varepsilon)^2\sigma^2(P'QR)} \\ &\leq t(u) + |up|\sqrt{\sigma(P'QR)^2 - \|\nabla\, t|_{qr}\|^2} \end{aligned}$$

which is precisely the causality constraint of Equation 3. The last inequality follows because PQR is progressive, hence $\|\nabla\, t|_{qr}\| \leq (1-\varepsilon)\sigma(P'QR)\phi_p \leq (1-\varepsilon)\sigma(P'QR)$.

Case 2: $\mathbf{t(u) < t(q)}$ See Figure 4(a). Let $\beta = |uq|/|up|$. Since $|uq| \neq 0$, we have

$$\begin{aligned} \frac{t'(p)-t(u)}{|up|} &= \frac{t'(p)-t(q)}{|up|} + \frac{t(q)-t(u)}{|uq|}\frac{|uq|}{|up|} \\ &= \frac{t'(p)-t(q)}{|up|} + \beta\|\nabla\, t|_{qr}\| \qquad (5) \end{aligned}$$

Using Equation 5, the causality constraint (Equation 3) can be rewritten as

$$\begin{aligned} \frac{t'(p)-t(q)}{|up|} \leq \sqrt{\sigma(P'QR)^2 - \|\nabla\, t|_{qr}\|^2} \\ - \beta\|\nabla\, t|_{qr}\| \qquad (6) \end{aligned}$$

Since t_i is progressive, we have $\|\nabla\, t|_{qr}\| \leq (1-\varepsilon)\sigma(P'QR)\phi_p$. Substituting this upper bound on $\|\nabla\, t|_{qr}\|$ into Equation 6, we obtain the following constraint:

$$\begin{aligned} \frac{t'(p)-t(q)}{|up|} \leq \sigma(P'QR)\left(\sqrt{1-(1-\varepsilon)^2\phi_p^2}\right) \\ - \sigma(P'QR)(1-\varepsilon)\beta\phi_p \qquad (7) \end{aligned}$$

which implies the causality constraint of Equation 6. Now,

$$\begin{aligned} \frac{t'(p)-t(q)}{|up|} &\leq \frac{t'(p)-t(p)}{|up|} \\ &\leq \frac{\varepsilon\sigma_{\min}w_{\min}}{|up|} \\ &\leq \varepsilon\sigma_{\min}. \end{aligned}$$

Since $\sigma_{\min} \leq \sigma(P'QR)$, Equation 7 is satisfied if

$$\varepsilon \leq \sqrt{1-(1-\varepsilon)^2\phi_p^2} - (1-\varepsilon)\beta\phi_p$$

or equivalently

$$(\varepsilon + (1-\varepsilon)\beta\phi_p)^2 + (1-\varepsilon)^2\phi_p^2 \leq 1$$

We have

$$(\varepsilon + (1-\varepsilon)\beta\phi_p)^2 + (1-\varepsilon)^2\phi_p^2 = 1 + 2\varepsilon(1-\varepsilon)\left(\beta\phi_p - 1\right)$$

We have $\phi_p = \sin\angle pqr = |up|/|pq| > |up|/|pr| = \sin\angle prq$ and $\beta = |uq|/|up|$. Since $|uq| < |pq|$, we have $\beta\phi_{qr} < 1$. Therefore, Equation 7 is satisfied. □

Theorem 5. *If a front t is progressive, then for any local minimum vertex p and for every $\delta t \in [0, T_{min}]$ the front $t' = next(t, p, \delta t)$ is progressive.*

Proof. Since only the triangles of the front incident on P advance along with p, we can restrict our attention to an arbitrary triangle pqr incident on p. Let t and t' denote $t_i|_{pqr}$ and $t_{i+1}|_{pqr}$ respectively. Let u be the orthogonal projection of p onto line qr. Let σ_{prog} denote $\sigma(P'Q'R)$ where $P' = (p, t(p) + \delta t)$ and $Q' = (q, t(q) + T_{\min})$.

We separate the analysis into three cases depending on which, if any, of the angles $\angle pqr$ and $\angle prq$ of $\triangle pqr$ is obtuse.

Case 1: Both $\angle$pqr and $\angle$prq are non-obtuse. See Figure 4(b). In this case, we have $t(u) \geq t(q) \geq t(p)$ and $\vec{n}_{qr} \cdot \vec{n}_{rp} \leq 0$. Let $\alpha = \sqrt{1-(\vec{n}_{qr}\cdot\vec{n}_{rp})^2}$. Hence, $\sin\angle qrp = \alpha = |up|/|pr|$ and $\vec{n}_{qr}\cdot\vec{n}_{rp} = -\sqrt{1-\alpha^2} = -|ur|/|pr|$. Also, $\phi_q = \max\{\sin\angle qrp, \sin\angle qpr\}$.

Therefore, the progress constraint of Equation 4 can be rewritten as follows:

$$\begin{aligned} \tfrac{t'(p)-t(u)}{|up|} \quad \leq \quad (1-\varepsilon)\sigma(P'Q'R)\tfrac{\max\{\sin\angle qrp, \sin\angle qpr\}}{\sin\angle qrp} \\ + \tfrac{|ur|}{|up|}\|\nabla\, t|_{qr}\| \\ (8) \end{aligned}$$

We have

$$\frac{t'(p)-t(u)}{|up|} \leq \frac{t'(p)-t(p)}{|up|} \leq \frac{\varepsilon\sigma_{\min}w_{\min}}{|up|} \leq \varepsilon\sigma_{\min}.$$

Since $\varepsilon \leq \frac{1}{2}$, we have $\varepsilon \leq 1-\varepsilon$; also, $\sigma_{\min} \leq \sigma(P'Q'R)$; hence,

$$\begin{aligned} \varepsilon\sigma_{\min} &\leq (1-\varepsilon)\sigma(P'Q'R) \\ &\leq (1-\varepsilon)\sigma(P'Q'R)\frac{\max\{\sin\angle qrp, \sin\angle qpr\}}{\sin\angle qrp} \\ &\leq (1-\varepsilon)\sigma(P'Q'R)\frac{\max\{\sin\angle qrp, \sin\angle qpr\}}{\sin\angle qrp} \\ &\quad + \frac{|ur|}{|up|}\|\nabla\, t|_{qr}\| \end{aligned}$$

Therefore, the progress constraint of Equation 8 is satisfied.

Case 2: $\angle$pqr is obtuse. See Figure 4(a). In this case, we have $t(u) < t(q)$ and $\vec{n}_{qr} \cdot \vec{n}_{rp} \leq 0$. Let $\alpha = \sqrt{1 - (\vec{n}_{qr} \cdot \vec{n}_{rp})^2}$. Hence, $\sin \angle qrp = \alpha = |up|/|pr|$ and $\vec{n}_{qr} \cdot \vec{n}_{rp} = -\sqrt{1-\alpha^2} = -|ur|/|pr|$.

Let $\beta = |uq|/|up|$. Since $|uq| \neq 0$, we have

$$\begin{aligned} \frac{t'(p) - t(u)}{|up|} &= \frac{t'(p) - t(q)}{|up|} + \frac{t(q) - t(u)}{|uq|}\frac{|uq|}{|up|} \\ &= \frac{t'(p) - t(q)}{|up|} + \beta \|\nabla\ t|_{qr}\| \end{aligned}$$

Therefore, the progress constraint of Equation 4 can be rewritten as follows:

$$\begin{aligned} \tfrac{t'(p)-t(q)}{|up|} \quad \leq \quad & (1-\varepsilon)\sigma(P'Q'R)\tfrac{\max\{\sin\angle qrp, \sin\angle qpr\}}{\sin\angle qrp} \\ & + \left(\tfrac{|ur|}{|up|} - \beta\right) \|\nabla\ t|_{qr}\| \end{aligned} \tag{9}$$

We have

$$\frac{t'(p) - t(q)}{|up|} \leq \frac{t'(p) - t(p)}{|up|} \leq \frac{\varepsilon\sigma_{\min}w_{\min}}{|up|} \leq \varepsilon\sigma_{\min}.$$

Since $\varepsilon \leq \frac{1}{2}$, we have $\varepsilon \leq 1-\varepsilon$; also, $\sigma_{\min} \leq \sigma(P'Q'R)$; hence,

$$\begin{aligned} \varepsilon\sigma_{\min} &\leq (1-\varepsilon)\sigma(P'Q'R) \\ &\leq (1-\varepsilon)\sigma(P'Q'R)\frac{\max\{\sin\angle qrp, \sin\angle qpr\}}{\sin\angle qrp} \\ &\leq (1-\varepsilon)\sigma(P'Q'R)\frac{\max\{\sin\angle qrp, \sin\angle qpr\}}{\sin\angle qrp} \\ &\quad + \left(\frac{|ur|}{|up|} - \beta\right) \|\nabla\ t|_{qr}\| \end{aligned}$$

Therefore, the progress constraint of Equation 9 is satisfied. The last inequality follows because $\beta = |uq|/|up| < |ur|/|up|$.

Case 3: $\angle$prq is obtuse. See Figure 4(c). In this case, we have $t(u) \geq t(r) \geq t(q) \geq t(p)$ and $\vec{n}_{qr} \cdot \vec{n}_{rp} > 0$. Let $\alpha = \sqrt{1 - (\vec{n}_{qr} \cdot \vec{n}_{rp})^2}$. Hence, $\sin\angle qrp = \alpha = |up|/|pr|$ and $\vec{n}_{qr} \cdot \vec{n}_{rp} = \sqrt{1-\alpha^2} = |ur|/|pr|$.

Let $\beta = |uq|/|up|$. Since $|uq| \neq 0$, we have

$$\begin{aligned} \frac{t'(p) - t(u)}{|up|} &= \frac{t'(p) - t(q)}{|up|} + \frac{t(q) - t(u)}{|uq|}\frac{|uq|}{|up|} \\ &= \frac{t'(p) - t(q)}{|up|} - \beta \|\nabla\ t|_{qr}\| \end{aligned}$$

Therefore, the progress constraint of Equation 4 can be rewritten as follows:

$$\begin{aligned} \tfrac{t'(p)-t(q)}{|up|} \quad \leq \quad & (1-\varepsilon)\sigma(P'Q'R)\tfrac{\max\{\sin\angle qrp, \sin\angle qpr\}}{\sin\angle qrp} \\ & + \left(\beta - \tfrac{|ur|}{|up|}\right) \|\nabla\ t|_{qr}\| \end{aligned} \tag{10}$$

As before, we have

$$\frac{t'(p) - t(q)}{|up|} \leq \frac{t'(p) - t(p)}{|up|} \leq \frac{\varepsilon\sigma_{\min}w_{\min}}{|up|} \leq \varepsilon\sigma_{\min}.$$

Since $\varepsilon \leq \frac{1}{2}$, we have $\varepsilon \leq 1-\varepsilon$; also, $\sigma_{\min} \leq \sigma(P'Q'R)$; hence,

$$\begin{aligned} \varepsilon\sigma_{\min} &\leq (1-\varepsilon)\sigma(P'Q'R) \\ &\leq (1-\varepsilon)\sigma(P'Q'R)\frac{\max\{\sin\angle qrp, \sin\angle qpr\}}{\sin\angle qrp} \\ &\leq (1-\varepsilon)\sigma(P'Q'R)\frac{\max\{\sin\angle qrp, \sin\angle qpr\}}{\sin\angle qrp} \\ &\quad + \left(\beta - \frac{|ur|}{|up|}\right) \|\nabla\ t|_{qr}\| \end{aligned}$$

Therefore, the progress constraint of Equation 10 is satisfied. The last inequality follows because $\beta = |uq|/|up| > |ur|/|up|$. □

Theorem 6. *For any $i \geq 0$, if the front t_i is progressive then t_i is valid.*

The proof is almost identical to that of Theorem 2.

3.3 Being greedy

We would like to maximize the progress at each step in a greedy fashion, i.e., given a front t_i we would like to maximize $t_{i+1}(p)$ where $t_{i+1} = \text{next}(t_i, p, \delta t)$ subject to the constraint that t_{i+1} is causal. For a fixed triangle $pp_1p_2\ldots p_d$ incident on p let T_{sup}^{i+1} denote $\sup\{T :$ $P'QR$ is causal and progressive, where $P' = (p, T)$ and $P_1' = (p_1, t_i(p_1) + T_{\min}\}$. To maximizing the progress at step $i+1$, we would like to compute T_{sup}^{i+1}. Similar to the 1D×Time case, partition the set of cones of influence from points on the front t_i into local and remote subsets. Let σ_{local} denote the smallest slope among all local cones of influence. The triangle $P'QR$ is causal only if its slope is less than or equal to σ_{local}. Let T_{local} be the maximum time value of P' for which the slope of $P'QR$ is less than or equal to σ_{local}. The maximum T_{local} exists because the set of allowed values of T where $P' = (p, T)$ is closed and therefore compact. To compute T_{local} we substitute σ_{local} in the condition for causality of $P'QR$.

Unlike the 1D×Time case, it is not clear that T_{sup}^{i+1} can be computed by ray shooting queries. In 2D×Time, we need an oracle to determine which among several right circular cones is intersected first by a triangle $P'Q'R$ when the vertex P of $\triangle PQR$ is lifted to $P' = (p, T)$ while also lifting Q to $Q' = (q, t(q) + T_{\min})$. However, just as for the 1D×Time case, we can approximate T_{sup}^{i+1} up to any given numerical accuracy by performing a binary search in the interval $[t_i(p) + T_{\min}, T_{\text{local}}]$ which we know contains T_{sup}^{i+1}. Therefore, the eventual height of the tentpole PP' is at least $\max\{T_{\min}, T_{\sup}^{i+1} - \eta\}$ where $\eta > 0$ is the desired numerical accuracy.

We thus have the following theorem.

Theorem 7. *Given a triangulation M of a bounded planar space domain where w_{min} is the minimum width of a simplex of M and σ_{min} is the minimum slope anywhere in $M \times [0, \infty)$, for every ε such that $0 < \varepsilon \leq \frac{1}{2}$ our algorithm constructs a simplicial mesh of $M \times [0, T]$ consisting of at most $\left\lceil \frac{diam(M)\,\sigma_{max}\,\Delta(M)}{\varepsilon\sigma_{min}w_{min}} T \right\rceil$ spacetime elements for every real $T \geq 0$.*

Proof. By Theorems 4 and 5, it follows that the height of each tentpole constructed by the algorithm is at least $T_{\min} = \varepsilon\sigma_{\min}w_{\min}$. By Theorem 6, after constructing at most $k \leq \left\lceil \frac{\mathrm{diam}(M)\,\sigma_{\max}}{T_{\min}} T \right\rceil$ patches, the entire front t_k is past the target time T. Since each patch consists of at most $\Delta(M)$ elements, the theorem follows. □

4. CONCLUSION

We have shown how to extend the Tent Pitcher algorithm for planar and linear spatial domains to the case of changing wavespeeds. Our expressions for the causality and progress constraints that apply at each step make explicit the dependence on the slope of the cone of influence most constraining the progress at that step. This dependence is not explicit in the formulæ of Erickson *et al.* because they assume without loss of generality that the slope is 1 everywhere in spacetime. For the constant wavespeed case, the algorithm in this paper is an alternative to the algorithm due to Erickson *et al.* with potentially weaker progress constraints. We can view the algorithm of Erickson *et al.* as looking one step ahead in the sense that the progress constraint at step i guarantees that the front constructed in step $i + 1$ is causal. Our algorithm can be viewed as looking one step even further—our progress constraint at step i guarantees that the front constructed in step $i + 2$ is causal. In a relatively straightforward manner, we can generalize this idea to looking at step i to the front in step $i + h$ where h is a *horizon* parameter that can be chosen adaptively by the algorithm. It needs to be investigated whether the extra complexity of the algorithm for $h > 2$ is justified by a more efficient meshing algorithm overall.

We have preliminary experimental results in 1D×Time and a prototype with simulated physics in 2D×Time; more substantial empirical study is required and we expect to report results of such a study soon. One of the objectives of the study will be to explore different heuristics to choose which local minimum vertex to pitch at every step. Some heuristics, such as pitching the local minimum with the minimum slope (highest wavespeed), perform better than others. We have an extension to the current algorithm that allows pitching at any vertex, not necessarily a local minimum. However, the extended algorithm is more complicated and it is not clear if the expected gains will be worth the extra computation time.

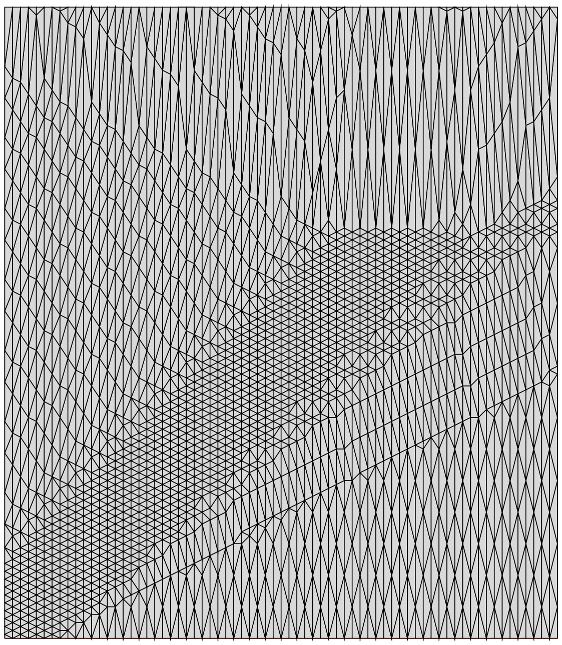

Figure 5: An unstructured triangular spacetime mesh over a 1D uniform space mesh. The space dimension is horizontal and time increases upwards. The slope at any point in spacetime is one of three distinct values: the minimum slope occurs in a band around the diagonal where the tentpoles are shortest; beyond a certain time value, the maximum slope occurs everywhere.

Figures 5 and 6 illustrate spacetime meshes constructed by our prototype implementation over 1D and 2D space meshes respectively. The 1D×Time spacetime mesh was constructed by pitching an independent set of local minima in non-increasing order of wavespeed. In other words, the algorithm preferred to pitch local minima adjacent to points on the front where the wavespeed was maximum (slope was minimum). The 2D×Time mesh was constructed by pitching a global minimum at every step. In either example, many more spacetime elements would be required to mesh the same volume if the height of every tentpole were constrained by the globally minimum slope.

In higher dimensions, we have a theorem identical to Theorem 7 when every dihedral angle of every simplex is non-obtuse. We anticipate soon an analogous theorem for arbitrary dimensional space domains in the presence of obtuse angles.

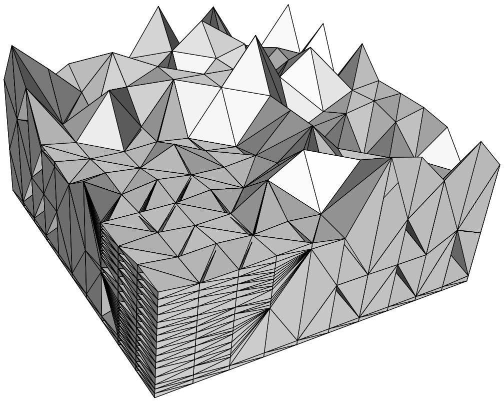

Figure 6: An unstructured tetrahedral spacetime mesh over a triangulated uniform 2D grid. Time increases upwards. The slope at any point in spacetime is one of two distinct values: the minimum slope occurs inside a circular cone where the tentpoles are shortest, the maximum slope occurs everywhere else.

Our algorithm can be modified to handle asymmetric cones, such as due to wave propagation through anisotropic media. In the presence of anisotropy, the most limiting cone constraint can be nonlocal.

In a recent paper, Abedi *et al.* [1] extend TentPitcher to support another kind of adaptivity, where the size of the spacetime elements is adapted to *a posteriori* estimates of the numerical error. Abedi *et al.* apply hierarchical refinement and coarsening of the underlying one- or two-dimensional space mesh to adapt the spatial size of future spacetime elements. They extend the progress constraints of Erickson *et al.* to anticipate future refinement and coarsening both of which change the shape of the elements on the front. The outstanding problem that we plan to consider next is to combine adaptivity to changing wavespeeds with refinement and coarsening for the case of planar space domains. It is quite straightforward to combine the progress constraints in this paper with those of Abedi *et al.* to support refinement in the presence of changing wavespeeds. Coarsening can be done safely if each triangle after coarsening satisfies progress constraint $[\sigma_{\min}]$. When coarsening is possible only under such strict constraints, we need to carefully prioritize each coarsening step so that the front is only as refined as necessary and not much more.

Our research group is also implementing a parallel version of Tent Pitcher to run on multiple processors. The nonlocal nature of the constraints pose significant challenges in the parallel setting.

In many problems, the geometry of the space domain changes over time. There may also be internal boundaries between different parts of the domain, e.g., separating two distinct materials with different physical properties, and these internal boundaries may evolve over time. We would like to handle moving boundaries both internal and external.

Acknowledgments The author would like to thank the other members of the CPSD spacetime meshing group, especially Jeff Erickson, Yong Fan, Robert Haber, Mark Hills, and Jayandran Palaniappan. Thanks also to the anonymous referee, whose comments were especially useful.

References

[1] Abedi R., Chung S.H., Erickson J., Fan Y., Garland M., Guoy D., Haber R., Sullivan J.M., Thite S., Zhou Y. “Spacetime Meshing with Adaptive Refinement and Coarsening.” *Proc. 20th Symp. Computational Geometry*, pp. 300–309. June 2004

[2] Barnes J.E., Hut P. “A Hierarchical $O(NLogN)$ Force Calculation Algorithm.” *Nature*, vol. 324, no. 4, 446–449, December 1986

[3] Cockburn B., Karniadakis G., Shu C. *Discontinuous Galerkin methods: theory, computation and applications*, vol. 11 of *Lecture Notes in Computational Science and Engineering*. Springer, 2000

[4] Erickson J., Guoy D., Sullivan J.M., Üngör A. “Building Space-Time Meshes over Arbitrary Spatial Domains.” *Proc. 11th Int'l. Meshing Roundtable*, pp. 391–402. 2002

[5] Guttman A. “A Dynamic Index Structure for Spatial Searching.” *Proc. ACM SIGMOD Conf. Principles Database Systems*, pp. 47–57. 1984

[6] Lin M., Manocha D., Cohen J., Gottschalk S. *Algorithms for Robotics Motion and Manipulation*, chap. Collision Detection: Algorithms and Applications, pp. 129–142. A.K. Peters, 1996

[7] Lowrie R.B., Roe P.L., van Leer B. “Space-Time Methods for Hyperbolic Conservation Laws.” *Barriers and Challenges in Computational Fluid Dynamics*, vol. 6 of *ICASE/LaRC Interdisciplinary Series in Science and Engineering*, pp. 79–98. Kluwer, 1998

[8] Richter G.R. “An explicit finite element method for the wave equation.” *Applied Numerical Mathematics*, vol. 16, 65–80, 1994

[9] Üngör A. *Parallel Delaunay Refinement and Space-Time Meshing*. Ph.D. thesis, University of Illinois at Urbana-Champaign, October 2002

[10] Üngör A., Sheffer A. “Tent-Pitcher: A Meshing Algorithm for Space-Time Discontinuous Galerkin Methods.” *Proc. 9th Int'l. Meshing Roundtable*, pp. 111–122. 2000

[11] Yin L., Acharya A., Sobh N., Haber R., Tortorelli D.A. “A Space-time discontinuous Galerkin method for elastodynamic analysis.” B. Cockburn, G. Karniadakis, C. Shu, editors, *Lecture Notes in Computational Science and Engineering*, vol. 11, pp. 459–464. Springer, 2000